

```
function MyEdgeDetector
%Pick an example and case
Example = 3;

if Example == 1
    fc = imread('3.gif');
    f0 = double(fc);
    thres = 0.35;
elseif Example == 2
    fc = imread('cameraman.png');
    f0 = double(rgb2gray(fc));
    thres = 0.25;
elseif Example==3
    fc = imread('winter-landscape.jpg');
    f0 = double(rgb2gray(fc));
    thres = 0.15;
end

[n, m] = size(f0);

%Filter Noise from Image
G = fspecial('gaussian',20,3);
f = imfilter(f0,G,'replicate');

%Compute Derivatives,
Dx = zeros(n,m);% df/dx
Dy = zeros(n,m);% df/dy
D = zeros(n,m);% |Df|

%Loop to compute derivatives and length of gradient
for j = 1:n-1
    for k = 1:m-1

        Dx(j,k) = f(j+1,k)-f(j,k);
        Dy(j,k) = f(j,k+1)-f(j,k);
        D(j,k) = sqrt((Dx(j,k)).^2+(Dy(j,k)).^2);

    end
end

%Rescale the norm of the gradient to be between 0 and 1
D = (D - min(D(:)))/(max(D(:)) - min(D(:)));

%Threshold the Norm of Gradient to find large jumps in image
TDf = double(D > thres);
```

```
%Thin the binary image, see bwmorph for help.  
Edge = bwmorph(TDf, 'Skel', inf);
```

```
%Plot Edges  
imagesc(Edge); colormap(gray);  
axis off; axis image; title('Edge')
```